SG12864-03D 系列模块说明书

(VERSION 1.0)

可选型号：

产品型号 及其后缀	LCD 类型 （显示模式）	背光类型 （LED）	时序方式	逻辑电压 （VDD）	背光 电压	接口方式 及其预留配置
SG12864FPD-03DSWE(3V)	FSTN 黑白正像	白背光	串行/并行 68	3.3V	3.3	单排插针焊位
SG12864FPD-03DSWE(5V)	FSTN 黑白正像	白背光	串行/并行 68	5.0V	5.0V	单排插针焊位
SG12864SBD-03DSWE(3V)	STN 蓝底色	白背光	串行/并行 68	3.3V	3.3V	单排插针焊位
SG12864SBD-03DSWE(5V)	STN 蓝底色	白背光	串行/并行 68	5.0V	5.0V	单排插针焊位
SG12864SYD-03DLYE(3V)	STN 黄绿底色	黄绿底背光	串行/并行 68	3.3V	5.0V	单排插针焊位
SG12864SYD-03DLYE(5V)	STN 黄绿底色	黄绿底背光	串行/并行 68	5.0V	5.0V	单排插针焊位
SG12864SGD-03DSWE(3V)	STN 灰底色	白背光	串行/并行 68	3.3V	3.3V	单排插针焊位
SG12864SGD-03DSWE(5V)	STN 灰底色	白背光	串行/并行 68	5.0V	5.0V	单排插针焊位
SG12864SYD-03DSYE(3V)	STN 黄绿底色	黄绿侧背光	串行/并行 68	3.3V	5.0V	单排插针焊位
SG12864SYD-03DSYE(5V)	STN 黄绿底色	黄绿侧背光	串行/并行 68	5.0V	5.0V	单排插针焊位

注：
以上列出 LCD 类型为我公司的标准品，如有其他需求，请与我公司销售部联系！

销售部 电话：010-51601528　　13683669085
QQ ：1056456564

文档修订记录

<table>
<tr><th rowspan="2">修订
次第</th><th rowspan="2">修订
日期</th><th rowspan="2">修订人</th><th rowspan="2">修订前
版本号</th><th colspan="3">修订</th><th rowspan="2">批准人</th></tr>
<tr><th>页
次</th><th>章节
编号</th><th>修订内容简述</th></tr>
<tr><td>1</td><td>2007-1-23</td><td>苏循刚</td><td>/</td><td>/</td><td>/</td><td>新建文档</td><td>趙鹏</td></tr>
</table>

目　　　录

1、显示模块整体描述 -- 4

2. 最大典型值--- 4

3. 电气特性--- 5

4. 光学特性--- 5

5. 光学特性测定方法--- 6

6. 原理框图--- 7

7. 时序图-- 7

8．功能说明及指令集 --11

9. LCD 驱动电源连接方式 ---21

10.液晶显示模块显示地址对应表 --22

11.出厂测试报告 --23

12. 接口引脚定义 ---24

13. 外形尺寸图纸 ---25

14.液晶显示模块在使用过程中的注意事项--27

15．硬件连接方式 ---29

16．驱动程序 --30

1、显示模块整体描述

项目	说明	单位
液晶显示模块组成	液晶显示屏,背光灯箱,线路板,铁框,导电橡胶等	
液晶显示屏类型	正像反射型,半透型,负像型	
液晶屏显示类型	STN 型:黄绿模式,灰模式,蓝模式	
	FSTN 型:黑白模式	
液晶显示屏视角	6 O'clock or 12 O'clock	
液晶模块外形尺寸 (LED*)	93.00(长)×70.00(宽)×13.00(厚)	mm
液晶模块视域	70.70(长)×38.80(宽)	mm
液晶模块铁框尺寸 (LED*)	80.50(长)×50.40(宽)×13.00(厚)	mm
液晶显示模块点阵数	128×64 点阵	
液晶显示屏点尺寸	0.48(长)×0.48(宽)	mm
液晶显示屏点间距	0.52(长)×0.52(宽)	mm
液晶显示屏占空比	1/32	
液晶显示屏偏置电压	1/5	
液晶显示模块控制器,驱动器	ST7920,ST7921 (COB)	
液晶显示模块使用温度范围(E*)	-20~+70	℃
液晶显示模块存储温度范围(E*)	-30~+80	℃
背光灯箱	LED: 黄绿底色、白色	
液晶显示模块数据输入格式	八位,四位并行输入格式,串行输入格式	
电源输入电压	单一 2.7V~5.5V 输入供电	V
液晶显示模块理论寿命	50,000	小时

注意: LED*: LED 背光

E*: 宽温等级

2. 最大典型值

2.1 电气最大典型值

Vss=0V

Item	Symbol	Min	Max	Unit	Note
逻辑电源	Vdd-Vss	0	7.0	V	
LCD 驱动电源	Vdd-Vo	0	Vdd	V	
I/O 输入电压	Vi	0	Vdd	V	

2.2 使用环境最大典型值

Item	Symbol	Min	Max	Unit
工作温度	T0	-20	+70	℃
储存温度	Ts	-30	+80	℃
湿度	---	---	85	%RH

3. 电气特性

3.1 电气特性(T_A=25℃,V_{DD}=2.7V-4.5V)

Item	Symbol	Condition	Min	Typ	Max	Unit
逻辑电源	Vdd	----	2.7	3.0	5.5	V
LCD 驱动电源	V_{LCD}	----	4.0	---	6.2	V
I/O 输入电压范围	VIH	----	0.7Vdd	---	Vdd	V
	VIL	----	-0.3	---	0.6	V
I/O 口输出电压范围	VOH	I_{OH}=-0.1mA	0.8Vdd	---	Vdd	V
	VOL	I_{OL}=0.1mA	--	---	0.1	V
各温度下,LCD 驱动电压范围	V_{LCD}	-20℃	5.4	5.8	6.2	V
	V_{LCD}	25℃	4.8	5.0	5.4	V
	V_{LCD}	70℃	4.0	4.4	4.8	V
频率	FOSC	Vdd=3.0V Rf=18KΩ	--	530K	--	Hz

3.2 LED 背光电气特性

Color	Item	Symbol	Min	Typ	Max	Unit	Condition
黄绿底背光	正向电压	Vf	4.0	4.2	4.4	V	If=360mA
黄绿侧背光	正向电压	Vf	3.7	3.9	4.1	V	If=120mA
白色侧背光	正向电压	Vf	2.8	3.0	3.2	V	If=65mA

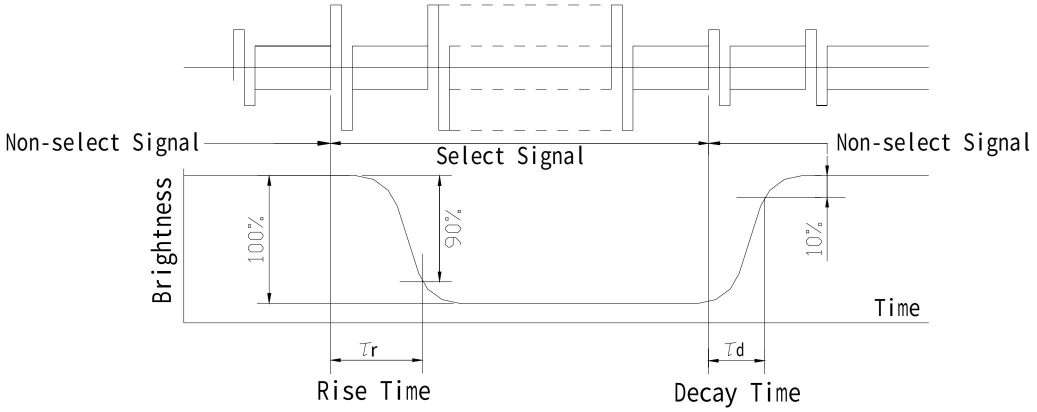
4. 光学特性

STN TYPE Ta=25℃

Item	Symbol	Condition	Min	Typ	Max	Unit	Reference
Viewing angle	θ	K≧2.0 φ=0o	40o	---	---	deg	
Contrast ration	K	θ=5o φ=0o	---	5	---	---	
Response time(rise)	Tr	θ=5o φ=0o	---	110	165	ms	
Response time(fall)	Tf	θ=5o φ=0o	---	110	165	ms	

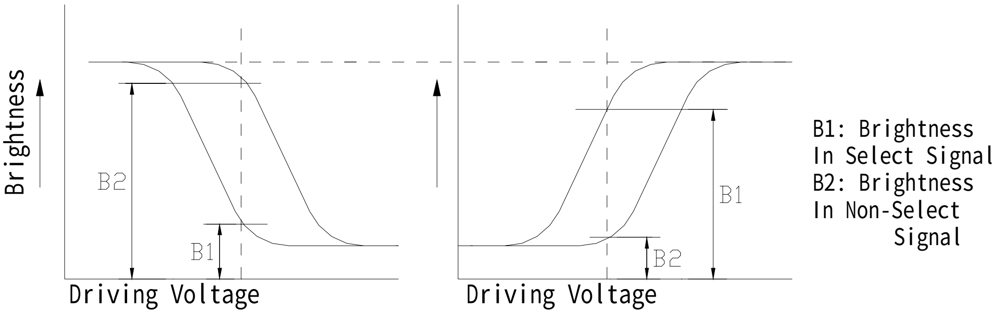
5. 光学特性测定方法

● Definition of Optical Response Time



In case of Negative type,
wave from of changing brightness becomes reverse
(Non Select Signals:0%,Select Signals:100%

● Definition of Contrast Ratio(CR)



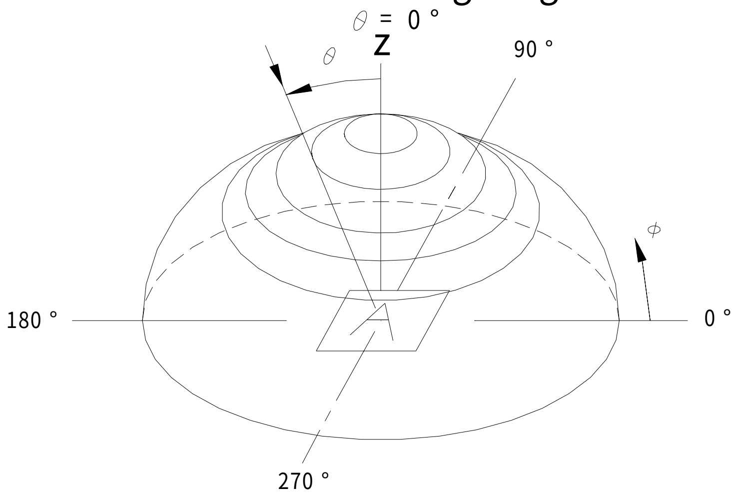
$$CR=\frac{\text{Brightness In Non-Select Signal}}{\text{Brightness In Select Signal}}$$

(Positive Type)

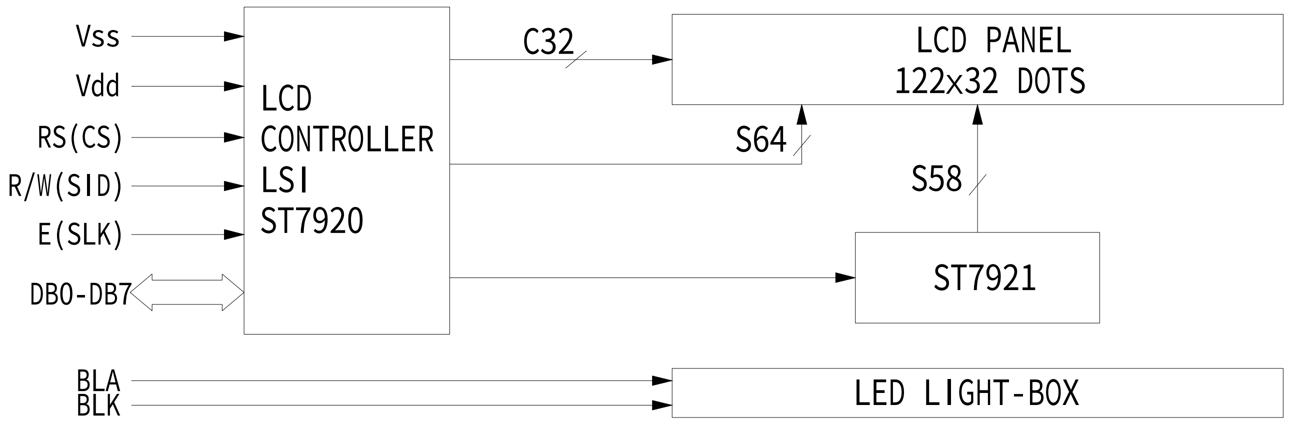
$$CR=\frac{\text{Brightness In Select Signal}}{\text{Brightness In Non-Select Signal}}$$

(Negative Type)

● Definition of Viewing Angle θ and φ



6. 原理框图



7. 时序图

7.1 并行方式 AC 特性(由 MPU 写资料到 ST7920) T_A=25℃,V_{DD}=4.5V

Characteristic	Symbol	Min	Typ	Max	Unit
Enable cycle time (Pin E)	T_C	1200	---	---	ns
Enable pulse width (Pin E)	T_{PW}	140	---	---	ns
Enable rise/fall time (Pin E)	T_R, T_F	---	---	25	ns
Address setup time (Pins RS, RW, E)	T_{AS}	10	---	---	ns
Address hold time (Pins RS, RW, E)	T_{AH}	20	---	---	ns
Data setup time (Pins DB0-DB7)	T_{DSW}	40	---	---	ns
Data hold time (Pins DB0-DB7)	T_H	20	---	---	ns

7.2 并行方式 AC 特性(由 ST7920 读资料到 MPU) T_A=25℃,V_{DD}=4.5V

Characteristic	Symbol	Min	Typ	Max	Unit
Enable cycle time (Pin E)	T_C	1200	---	---	ns
Enable pulse width (Pin E)	T_{PW}	140	---	---	ns
Enable rise/fall time (Pin E)	T_R, T_F	---	---	25	ns
Address setup time (Pins RS, RW, E)	T_{AS}	10	---	---	ns
Address hold time (Pins RS, RW, E)	T_{AH}	20	---	---	ns
Data delay time (Pins DB0-DB7)	T_{DDR}	---	---	100	ns
Data hold time (Pins DB0-DB7)	T_H	20	---	---	ns

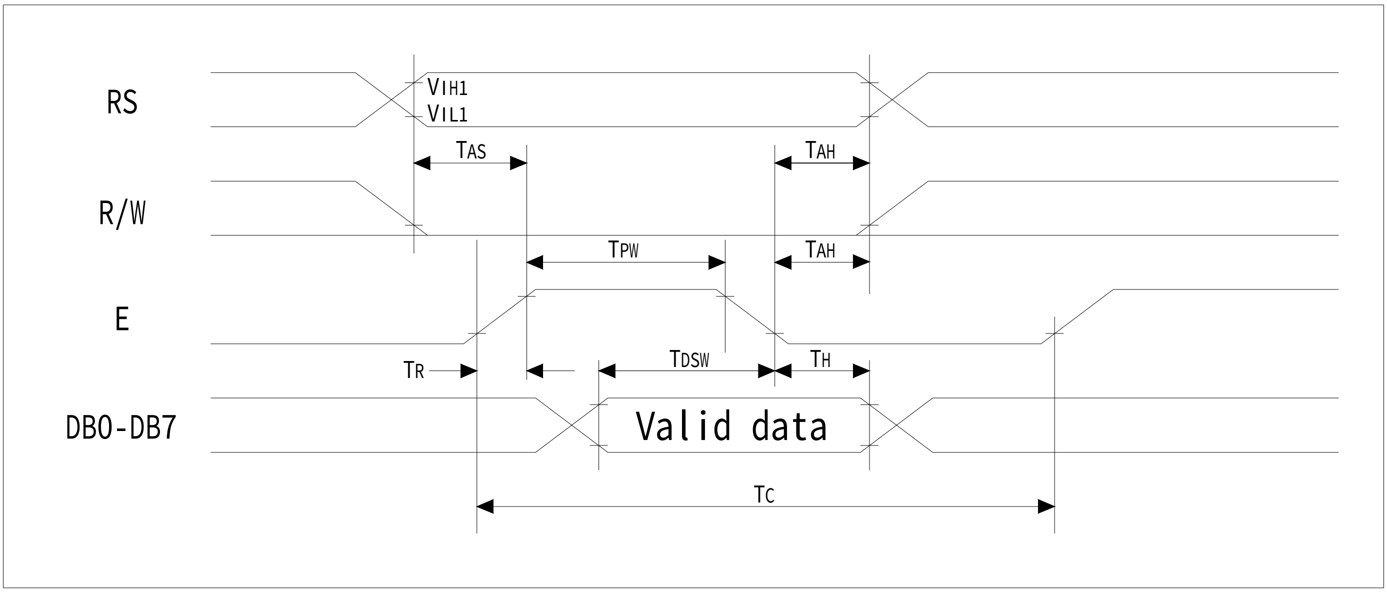
7.3 并行方式 AC 特性(由 MPU 写资料到 ST7920) T_A=25℃,V_{DD}=2.7V

Characteristic	Symbol	Min	Typ	Max	Unit
Enable cycle time (Pin E)	T_C	1800	---	---	ns
Enable pulse width (Pin E)	T_{PW}	160	---	---	ns
Enable rise/fall time (Pin E)	T_R, T_F	---	---	25	ns
Address setup time (Pins RS, RW, E)	T_{AS}	10	---	---	ns
Address hold time (Pins RS, RW, E)	T_{AH}	20	---	---	ns
Data setup time (Pins DB0-DB7)	T_{DSW}	40	---	---	ns
Data hold time (Pins DB0-DB7)	T_H	20	---	---	ns

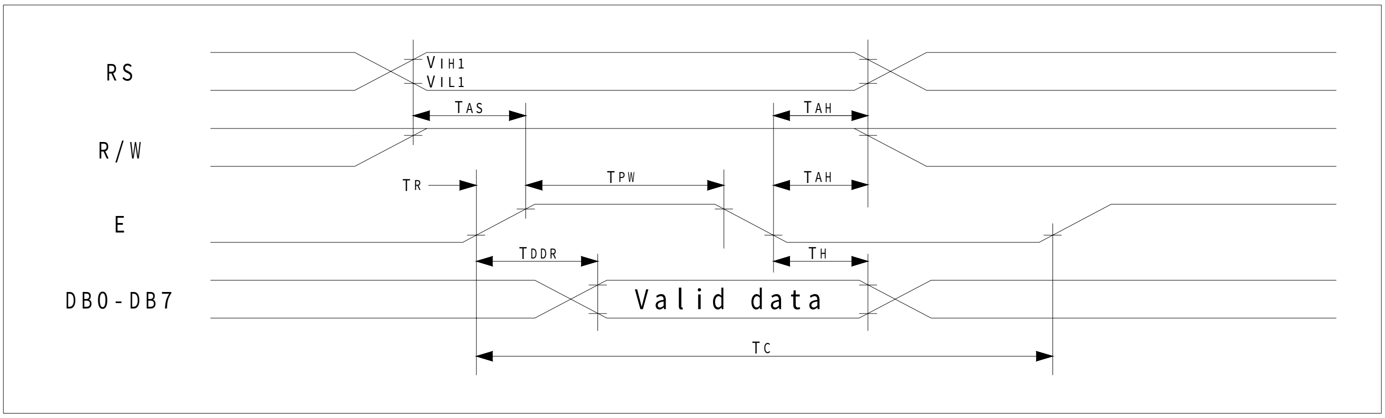
7.4 并行方式 AC 特性(由 ST7920 读资料到 MPU) T_A=25℃,V_{DD}=2.7V

Characteristic	Symbol	Min	Typ	Max	Unit
Enable cycle time (Pin E)	T_C	1800	---	---	ns
Enable pulse width (Pin E)	T_{PW}	320	---	---	ns
Enable rise/fall time (Pin E)	T_R, T_F	---	---	25	ns
Address setup time (Pins RS, RW, E)	T_{AS}	10	---	---	ns
Address hold time (Pins RS, RW, E)	T_{AH}	20	---	---	ns
Data delay time (Pins DB0-DB7)	T_{DDR}	---	---	260	ns
Data hold time (Pins DB0-DB7)	T_H	20	---	---	ns

7.5 由 MPU 写资料到 ST7920



7.6 由 ST7920 读资料到 MPU

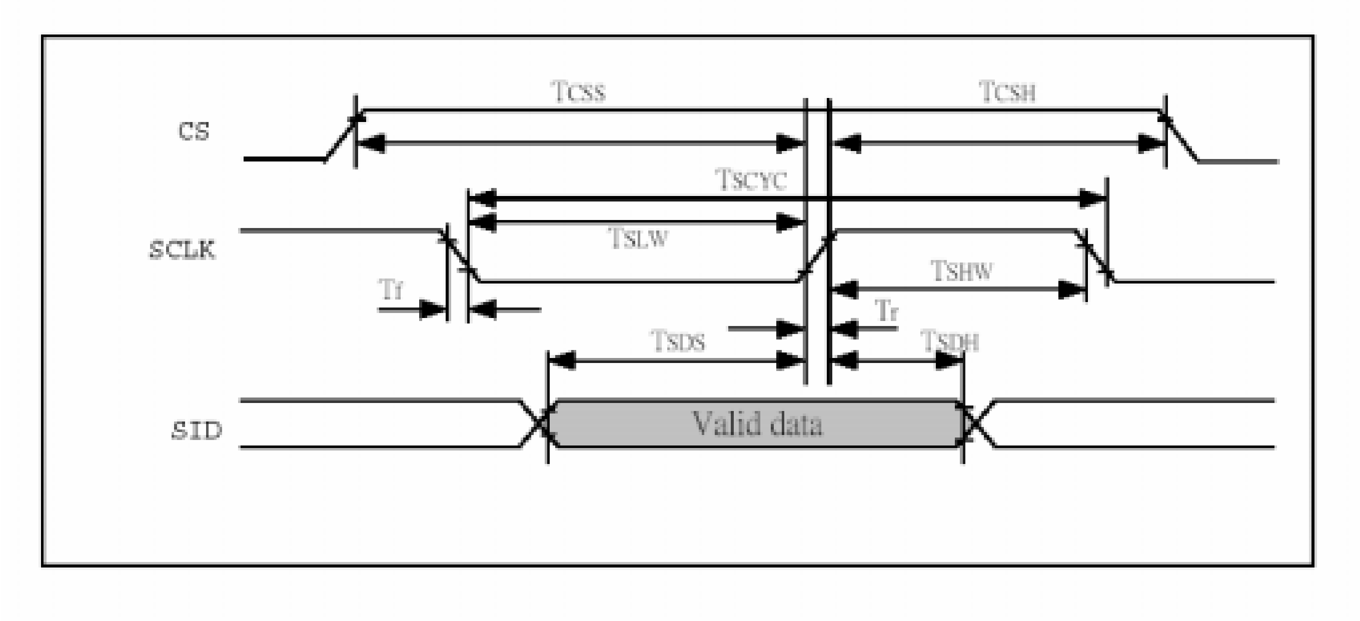


7.7 串行方式 AC 特性 T_A=25℃,V_{DD}=4.5V

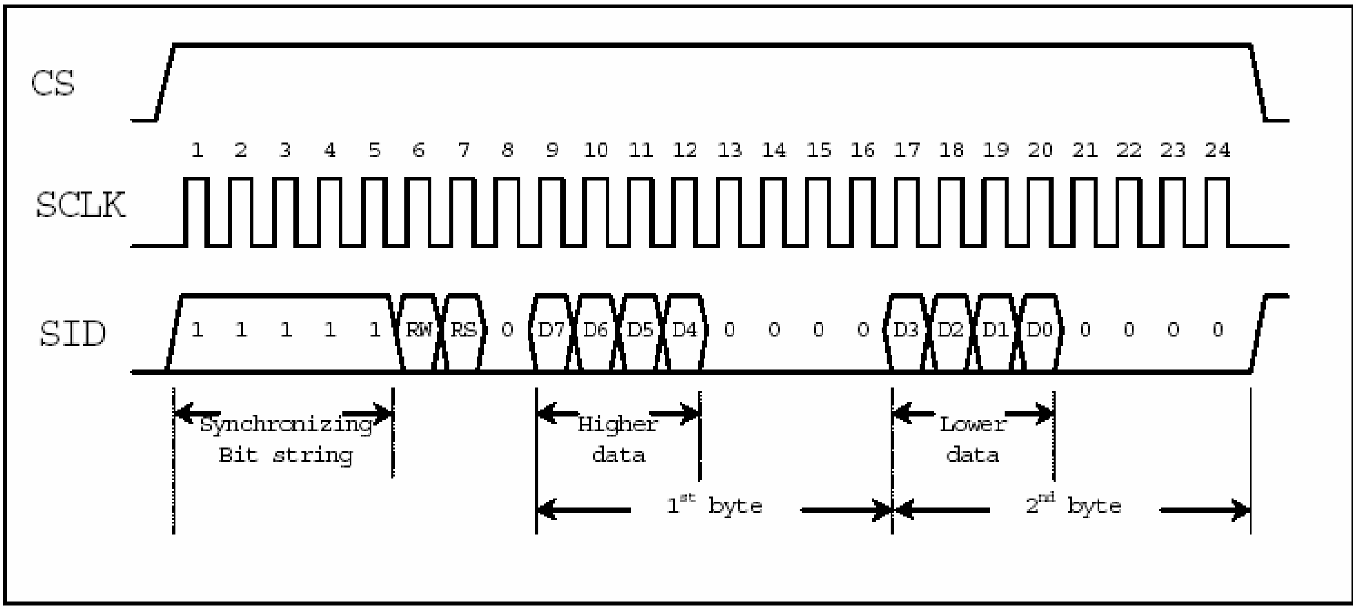
Characteristic	Symbol	Min	Typ	Max	Unit
Rise/Fall time	T_R, T_F	---	---	02.	μs
Serial clock time (Pin E)	T_{SCYC}	400	---	---	ns
SCLK high pulse width (Pin E)	T_{SHW}	200	---	---	ns
SCLK low pulse width (Pin E)	T_{SLW}	200	---	---	ns
SID data setup time (Pins RW)	T_{SDS}	40	---	---	ns
SID data hold time (Pins RW)	T_{SDH}	40	---	---	ns
CS time (Pins RS)	T_{CSS}	60	---	---	ns
CS hold time (Pins RS)	T_{CSH}	60	---	---	ns

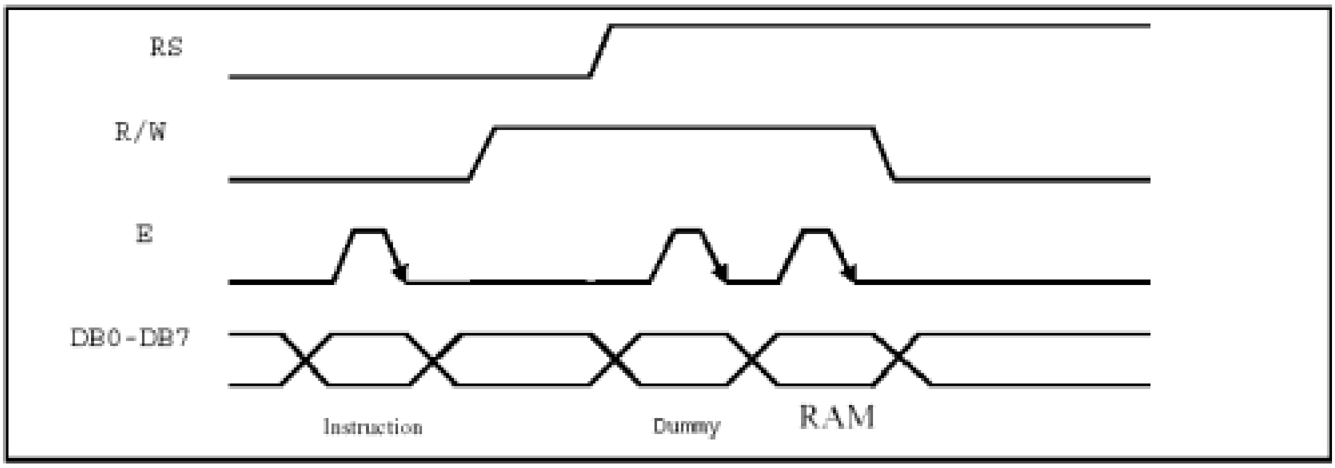
7.8 串行方式 AC 特性 T_A=25℃,V_{DD}=2.7V

Characteristic	Symbol	Min	Typ	Max	Unit
Rise/Fall time	T_R, T_F	---	---	02.	μs
Serial clock time (Pin E)	T_{SCYC}	600	---	---	ns
SCLK high pulse width (Pin E)	T_{SHW}	300	---	---	ns
SCLK low pulse width (Pin E)	T_{SLW}	300	---	---	ns
SID data setup time (Pins RW)	T_{SDS}	40	---	---	ns
SID data hold time (Pins RW)	T_{SDH}	40	---	---	ns
CS time (Pins RS)	T_{CSS}	60	---	---	ns
CS hold time (Pins RS)	T_{CSH}	60	---	---	ns

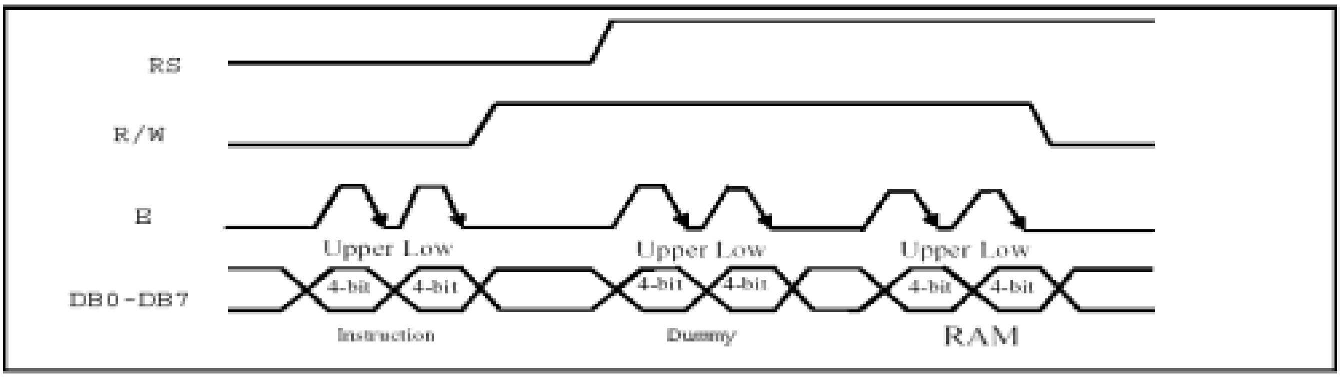


7.9 各种接口方式时序图





Timing Diagram of 8-bit Parallel Bus Mode Data Transfer



Timing Diagram of 4-bit Parallel Bus Mode Data Transfer

8. 功能说明及指令集

8.1. 系统功能

8.1-1 功能描述

本模块使用 ST7920 控制器,并配合使用 ST7921 驱动器来形成产品,其中所有的操作针对 ST7920 完成,ST7920 可以提供三种方式来连接微处理器:

①8 位并行方式
②4 位并行方式
③串行方式

本模块为 8 位并行方式

在读或写操作到 ST7920 中,有两个 8 位暂存器将会使用到,一个是数据暂存器(DR),另一个是指令暂存器(IR),通过数据暂存器(DR)可以存取 DDRAM/CGRAM/GDRAM 以及 IRAM 的值,待存取目标 RAM 的地址,通过指令来选择,每次数据暂存器(DR)的存取操作都将以上次选择的目标 RAM 地址为主体来做写入或读出.

配合 RS 及 RW 可以决定控制方式的 4 种读写模式,如下表:

RS	R/W	Function
0	0	MPU 写指令到指令暂存器(IR)
0	1	读忙标志(BF)及地址计数器(AC)的状态
1	0	MPU 写入资料到数据暂存器
1	1	MPU 从数据暂存器(DR)中读出数据

8.1-2 忙标志(BF)

当 BF 为“1”时,表示内部操作正在进行中处于忙状态,此时不能接受新的指令操作,要输入新的指令前,必须先读取 BF 标志,直到 BF 标志为“0”时,才能接受新的指令注入,一般而言任何的指令注入后 ST7920 内部都需要时间处理,在处理完成前并不能接受下一个指令,而每一个指令的处理时间并不相同,所以要知道 ST7920 内部是否可以接受下一指令,需要由读取 BF 标志为来确认.

8.1-3 地址计数器(AC)

地址计数器(AC)用于存储 DDRAM/CGRAM/IRAM/GDRAM 之一的地址,它可以由设定指令暂存器(IR)来改变,在此之后只要读取或写入 DDRAM/CGRAM/IRAM/GDRAM 的值时,地址计数器(AC)的值就会自动增一,当 RS 为“0”时而 RW 为“1”时,地址计数器(AC)的值会被读取到 DB6-DB0 中.

8.1-4 中文字型生成 ROM(CGROM)及半宽字型 ROM(HCGROM)

ST7920 字型生成 ROM 中可以提供 8192 个 16×16 点阵的中文字型以及 126 个 16×8 点阵的数字字型,它使用两个字节来组成字型编码选择,配合 DDRAM 将要显示的字模码写入到 DDRAM 上,字体将自动的依照编码从 CGROM 中显示在 LCD 上.

8.1-5 字型产生 RAM(CGRAM)

ST7920 字型产生 RAM 提供给使用者自定义(造字)功能,可以提供四组 16×16 点阵的自定义空间,使用者可以将内部字模中没有的字型自行定义到 CGRAM 中,通过 DDRAM 显示在 LCD 上.

8.1-6 ICON RAM (IRAM)

ST7920 提供 240 点的 ICON 显示,它分别由 15 组的 IRAM 地址来组成,每一组 IRAM 地址由 16 个字节构成,每次写入一组 IRAM 时,需先指定 IRAM 的地址,再通过连续写入两个字节的数据完成,先写入高字节(D15--D8)再写入低字节(D7--D0).

8.1-7 显示数据 RAM(DDRAM)

显示数据 RAM 可以提供 64×2 个字节的空间,最多可以控制 4 行 16 个字(64 个字)的中文字型显示,当写入显示数据 RAM 时,可以分别显示 CGROM,HCGROM 与 CGRAM 的字型.ST7920 可以显示三种字型,分别是半宽的 HCGROM 字型,CGRAM 字型及中文 CGROM 字型,三种字型的选择,由在 DDRAM 中写入的编码选择,在 0000H-0006H 的编码中将选择 CGRAM 的自定义字型,02H-07FH 的编码中将选择半宽英文,数字的字型,至于 A1 以上的编码将自动的结合下一个字节,组成两个字节的编码达到中文字型的编码 GB(A1A0-F7FF),各种详细字型编码如下:

1. 显示半宽字型:将 8 位数据写入 DDRAM 中,范围在 02H-07H 的编码.
2. 显示 CGRAM 字型:将 16 位数据写入 DDRAM 中,共有 0000H,0002H,0004H,0006H 四种编码.
3. 显示中文字型:将 16 位数据写入 DDRAM 中范围在:A1A0H-F7FFH 的编码.将 16 位数据写入 DDRAM 方式为通过连续写入 2 个字节的数据来完成,先写入高字节(D15-D8)再写入低字节(D7-D0).

CGRAM 字型与中文字型编码只可以出现在每一 Address counter 的起始位置
具体请参考本资料中第 10 项说明.

8.1-8 图形显示 RAM(GDRAM)

图形显示 RAM 提供 64×32 个字节的存储空间(由扩充指令设定图形显示 RAM 地址),最多可以控制 256×64 点的图形空间,在更改图形显示 RAM 时,由扩充指令设定 GDRAM 的地址,先设置垂直地址,再设置水平地址(连续写入两个字节的数据来完成垂直和水平的坐标位置),再写入两个字节的数据到图形显示 RAM,而地址计数器(AC)会自动增一,写入图形显示 RAM 的步骤如下:

1. 先将垂直的坐标(Y)写入图形显示 RAM 地址.
2. 再将水平的坐标(X)写入图形显示 RAM 地址.
3. 将 D15-D8 写入到 RAM 中(写入第一个 Bytes).
4. 将 D7-D0 写入到 RAM 中(写入第二个 Bytes).

具体请参考本资料中第 10 项说明

8.1-9 光标/闪烁

ST7920 提供光标及闪烁功能,由地址计数器(address counter)的值来指定 DDRAM 中的光标或闪烁位置.

8.2 指令集

ST7920 共有两套控制指令,分别为基本指令和扩充指令,下面做详细说明.

8.2-1 基本指令集(RE=0)

指令	指令码										说明	执行时间
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		(540KHZ)
清除显示	0	0	0	0	0	0	0	0	0	1	将 DDRAM 填满 20H,并设定 DDRAM 的地址计数器(AC)到 00H	1.6ms
地址归位	0	0	0	0	0	0	0	0	1	X	设定 DDRAM 的地址计数器(AC)到"00H",并且将游标移到开头原点位置;这个指令并不改变 DDRAM 的内容	72μs
进入点设定	0	0	0	0	0	0	0	1	I/D	S	设定在数据的读取与写入时,设定光标移动方向及指定显示的移位	72μs
显示状态开/关	0	0	0	0	0	0	1	D	C	B	D=1：整体显示 ON C=1：光标 ON B=1：光标位置 ON	72us
光标或显示移位控制	0	0	0	0	0	1	S/C	R/L	X	X	设定光标的移动与显示的移位控制;这个指令并不改变 DDRAM 的内容	72us
功能设定	0	0	0	0	1	DL	X	0 RE	X	X	DL=1 （必须设为 1） RE=1： 扩充指令集动作 RE=0： 基本指令集动作	72us
设定 CGRAM 地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	设定 CGRAM 地址到地址计数器(AC),需确认扩充指令中 SR=0(卷动地址或 RAM 地址选择)	72us
设定 DDRAM 地址	0	0	1	0 AC6	AC5	AC4	AC3	AC2	AC1	AC0	设定 DDRAM 地址到地址计数器(AC),AC6 固定为 0	72us
读忙标志和地址	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	读取忙标志(BF)可以确认内部动作是否完成,同时可以读出地址计数器(AC)的值	0us
写数据到 RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	写入数据到内部的 RAM(DDRAM/CGRAM/IRAM/GDRAM)	72us
读出 RAM 的值	1	1	D7	D6	D5	D4	D3	D2	D1	D0	从内部 RAM 读出资料(DDRAM/CGRAM/IRAM/GDRAM)	72us

基本指令集说明:

● 清除显示

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

将 DDRAM 填满“ 20H ”,并且设定 DDRAM 的地址计数器(AC)到“ 00H ”,重新进入点设定将 I/D 设为“ 1 ”光标右移增 1.

● 地址归位

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	X

设定 DDRAM 的地址计数器(AC)到“ 00H ”,并且将光标移到起始原点位置,这个指令并不改变 DDRAM 的内容.

● 进入点设定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

指定在数据的读取与写入时,设定光标的移动方向及指定显示的移位.

I/D:地址计数器递增递减选择

I/D=1,光标右移,DDRAM 地址计数器(AC)增 1.

I/D=0,光标左移,DDRAM 地址计数器(AC)减 1.

S:显示画面整体移动

S	I/D	DESCRIPTION
H	H	画面整体左移
H	L	画面整体右移

● 显示状态开/关

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

控制整体显示,光标,光标位置反白 ON/OFF

D:整体显示 ON/OFF

D=1, 整体显示 ON.

D=0, 整体显示 OFF,但不改变 DDRAM 的内容.

C:光标 ON/OFF 控制

C=1, 光标显示 ON.

C=0, 光标显示 OFF.

B:光标位置反白 ON/OFF 控制

B=1, 光标位置显示反白 ON,将光标所在地址上的资料反白显示.

B=0, 光标位置显示反白 OFF.

- 光标或显示移动控制

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

设定光标的移动与显示的移动控制,这个指令并不改变 DDRAM 的内容.

S/C	R/L	DESCRIPTION	AC VALUE
L	L	光标向左移动	AC=AC-1
L	H	光标向右移动	AC=AC+1
H	L	显示向左移动,且光标跟随移动	AC=AC
H	H	显示向右移动，且光标跟随移动	AC=AC

- 功能设定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	X	RE	X	X

DL:4/8BIT 操作控制

DL=1，为 8BIT MPU 操作.

DL=0，为 4BIT MPU 操作.

RE:指令集选择控制

RE=1，为选择扩充指令集操作.

RE=0，为选择基本指令集操作.

同一指令的操作不能同时改变 RE 及 DL,需要先改变 DL 后,再改变 RE 才可以取保 FLAG 正确设定.

- 设定 CGRAM 地址

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

设定 CGRAM 地址到地址计数器(AC),AC 范围:00H—3FH..

需确认扩充指令中 SR=0(卷动地址或 RAM 地址选择).

- 设定 DDRAM 地址

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

设定 DDRAM 地址到地址计数器(AC).

第一行 AC 范围:80H—87H

第二行 AC 范围:90H—97H

第三行 AC 范围:88H—8FH

第四行 AC 范围:98H—9FH

● 读取忙标志位(BF)和地址

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

读取忙标志位(BF)可以确认内部工作是否完成,同时可以读出地址计数器(AC) 的值,当 BF=?? 表示内部忙碌中,此时不能下达新的指令,需等待 BF=?才可以下达新的指令.

● 写入数据到 RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

写入数据到内部的 RAM,当写入后会改变 AC.
每个 RAM 地址(CGRAM,DDRAM,IRAM)都可以连续写入两个字节的数据(2-Bytes),当写入第二 BYTE 时地址计数器(AC)的值会自动增一.

● 读取 RAM 的值

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

从内部的 RAM 读取数据,当读取后会使 AC 改变.
在设定地址指令后(CGRAM,DDRAM,IRAM),若要读取数据时先 DUMMY READ 一次才会读取到正确的数据,第二次读取时侧不需要 DUMMY READ,除非又设定地址指令后才需在次 DUMMY READ.

8.2-2 扩充指令集(RE=1)

指令	指令码										说明	执行时间 (540KHZ)
	RS	**R/W**	**DB7**	**DB6**	**DB5**	**DB4**	**DB3**	**DB2**	**DB1**	**DB0**		
待命模式	0	0	0	0	0	0	0	0	0	1	将 DDRAM 填满 20H,并设定 DDRAM 的地址计数器(AC)到 00H	72μs
卷动地址或 IRAM 地址选择	0	0	0	0	0	0	0	0	1	SR	SR=1：允许输入垂直卷动地址 SR=0：允许输入 IRAM 地址	72μs
反白模式	0	0	0	0	0	0	0	1	R1	R0	选择 4 行中的任一行作反白显示，并可决定反白与否,R1,R0 初始为 00 当第一次设定时为反白显示,再一次设定时为正常显示	72μs
睡眠模式	0	0	0	0	0	0	1	SL	X	X	SL=1：脱离睡眠模式 SL=0：进入睡眠模式	72us
扩充功能设定	0	0	0	0	1	DL	X	1 RE	G	0	DL=1 8-BIT 控制模式 DL=0 4-BIT 控制模式 E=1： 扩充指令集动作	72us

											RE=0： 基本指令集动作 G=1 ：图形显示 ON G=0 ：图形显示 OFF	
设定 IRAM 地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	SR=1 ：AC5—AC0 为垂直卷动地址 SR=0：AC3—AC0 为 ICON IRAM 地址	72us
设定 图形显示 RAM 地址	0	0	1	0 AC6	0 AC5	0 AC4	AC3 AC3	AC2 AC2	AC1 AC1	AC0 AC0	设定 CGRAM 地址到地址计数器(AC),先设定垂直地址再设定水平地址(连续写入两个字节的数据来完成垂直与水平坐标地址) 垂直地址范围:AC6—AC0 水平地址范围:AC3—AC0	72us

备注：

1.当模块在接受指令前，微处理器必须先确认模块内部处于非忙碌状态,即读取 BF 标志时 BF 需为 0,方可接受新的指令.如果在送出一个指令前并不检查 BF 标志,那么在前一个指令和这个指令中间必须延迟一段较长的时间,即是等待前一个指令确实执行完成,指令执行的时间请参考指令表中的个别指令说明。

2.“ RE ”为基本指令集与扩充指令集的选择控制,当变更“ RE ”后,往后的指令集将维持在最后的状态,除非再次变更“ RE ”,否则使用相同指令集时,不需每次重设“ RE ”.

扩充指令集说明

● 待命模式

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

进入待命模式,执行任何的其它指令都可以终止待命模式,这条指令并不改变 RAM 的内容.

●卷动地址或 RAM 地址选择

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	SR

SR=1，允许输入垂直卷动地址.

SR=0，允许输入 IRAM 地址(填充指令)及允许设定 CGRAM 地址(基本指令).

● 反白模式

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	R1	R0

选择四行中的任一行作反白显示,并可以决定是否反白.

R1,R0 初始值为 00,当第一次设定后为反白显示,再一次设定后为正常显示:

R1	R0	DESCRIPTION
L	L	第一行反白或正常显示
L	H	第二行反白或正常显示
H	L	第三行反白或正常显示
H	H	第四行反白或正常显示

● **睡眠模式**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	SL	0	0

SL=1，脱离睡眠模式.

SL=0，进入睡眠模式.

● **扩充功能设定**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	X	RE	G	X

DL:4/8BIT 操作控制

DL=1，为 8BIT MPU 操作.

DL=0，为 4BIT MPU 操作.

RE:指令集选择控制

RE=1，为选择扩充指令集操作.

RE=0，为选择基本指令集操作.

G:图形显示控制

G=1，图形显示 ON.

G=0，图形显示 OFF.

同一指令的操作不能同时改变 RE,DL,G,需先改变 DL 或 G 后,再改变 RE,才可以确保 FLAG 正确设定.

● **设定 IRAM 地址或卷动地址**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

SR=1，AC5--AC0 为垂直卷动地址.

SR=0，AC3--AC0 为 ICON RAM 地址.

● **设定图形显示 RAM 地址**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

设定 GDRAM 地址到地址计数器(AC)

先设垂直地址再设水平地址(连续写入两个字节的资料来完成垂直和水平的坐标位置)

垂直地址范围:AC6—AC0.

水平地址范围:AC3—AC0.

图形显示 RAM 的地址计数器(AC),只会对水平地址(X 轴)自动增一,当水平地址=0FH 时会重新设为 00H,但并不会对垂直地址做仅为自动增一,所以当连续写入多个位置显示方式时,需要自行判断垂直地址是否需要重新设定.

8.2-3 基本指令集初始值(Register flag)(RE=0)

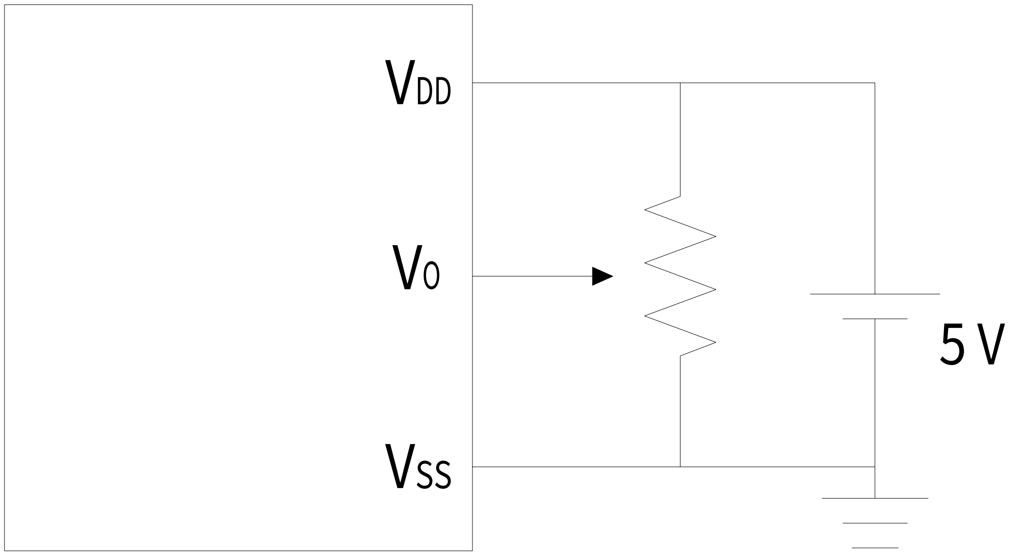
指令	指令码										说明
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
进入点设定	0	0	0	0	0	0	0	1	I/D	S	光标右移,DDRAM 地址计数器(AC)自动增 1
									1	0	
显示状态开/关	0	0	0	0	0	0	1	D	C	B	控制整体显示,光标,光标位置反白 ALL OFF
								0	0	0	
光标或显示移位控制	0	0	0	0	0	1	S/C	R/L	X	X	无光标与显示一位动作
							X	X			
功能设定	0	0	0	0	1	DL	X	0 RE	X	X	8 BIT MPU 控制界面,基本指令动作
						1		0			

8.2-4 扩充指令集初始值(Register flag)(RE=1)

指令	指令码										说明
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
卷动地址或 IRAM 地址选择	0	0	0	0	0	0	0	0	1	SR	允许输入 IRAM 地址或设定 CGRAM 地址
										0	
反白模式	0	0	0	0	0	0	0	1	R1	R0	当第一次设定时为反白显示,再一次设定时为正常显示.
									0	0	
睡眠模式	0	0	0	0	0	0	1	SL	X	X	未进入待命模式
								0			
扩充功能设定	0	0	0	0	1	DL	X	1 RE	G	0	图形显示 OFF
								0			

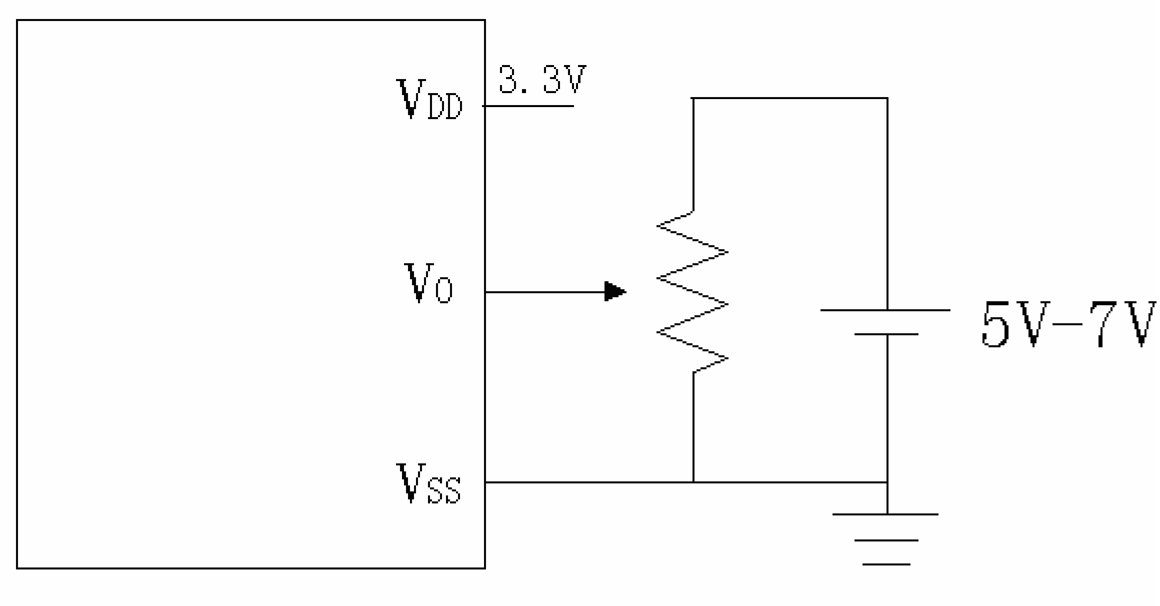
9. LCD 驱动电源连接方式

后缀为 5V 的模块



VR=10K~20K

后缀为 3V 的模块



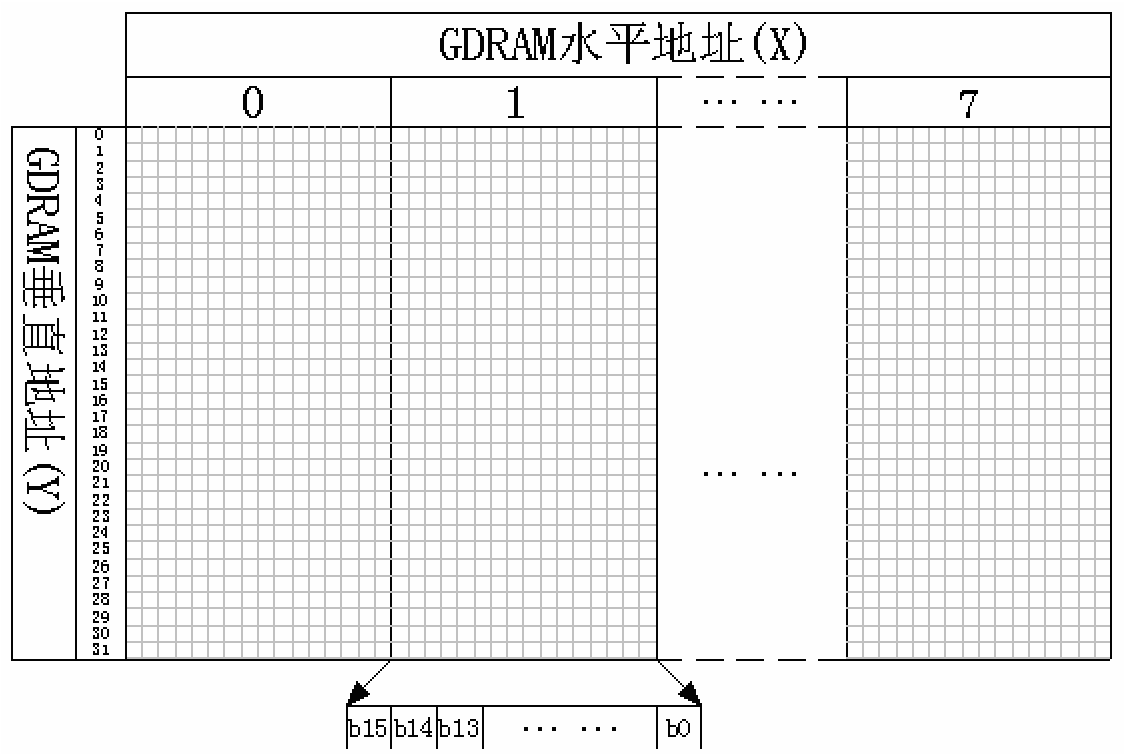
VR=10K~20K

10.液晶显示模块显示地址对应表

10.1 中文显示 RAM 地址

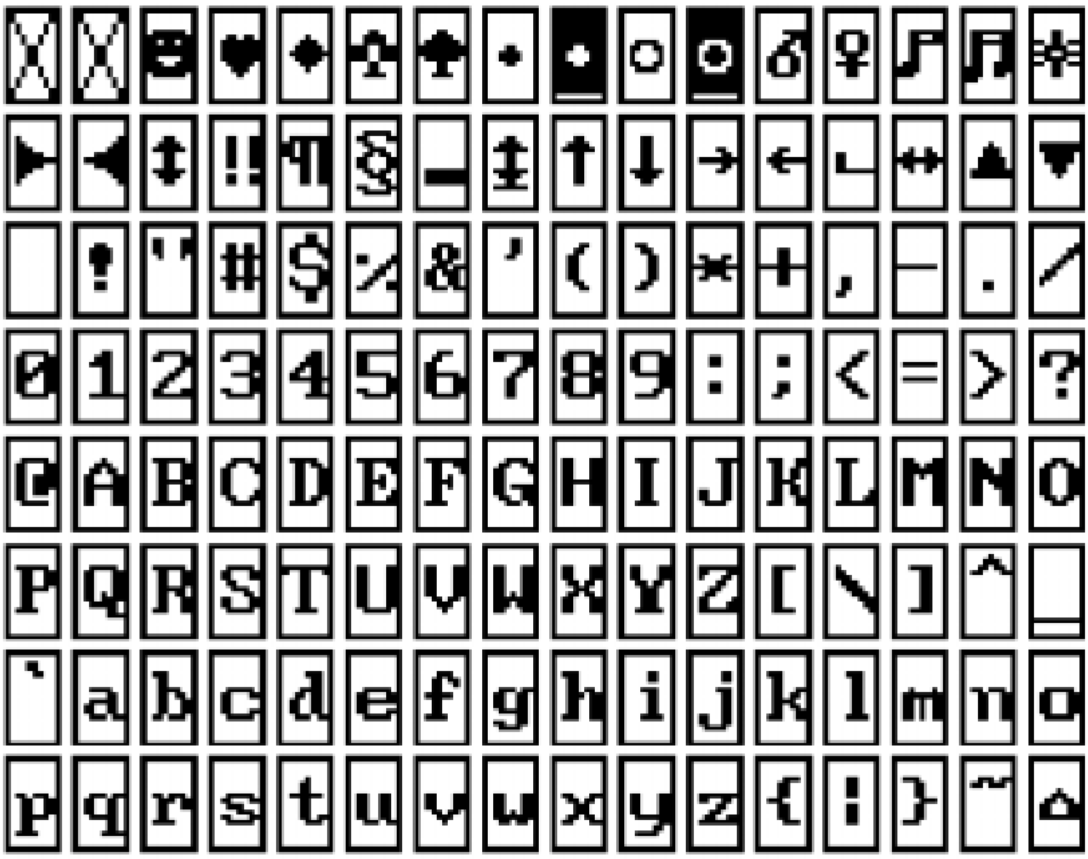
字型	1	2	3	4	5	6	7	8
中文显示第一行	80H	81H	82H	83H	84H	85H	86H	87H
中文显示第二行	90H	91H	92H	93H	94H	95H	96H	97H
中文显示第二行	88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
中文显示第二行	98H	98H	9AH	9BH	9CH	9DH	9EH	9FH

10.2 图形显示方式坐标



10.3 控制器中内藏的字符表(16×8 半宽字型)

控制代码（02H---7FH）



10.4 中文字库代码请参考 ST7920 资料.

11.出厂测试报告

V_{DD}=5V　　Ta=25℃

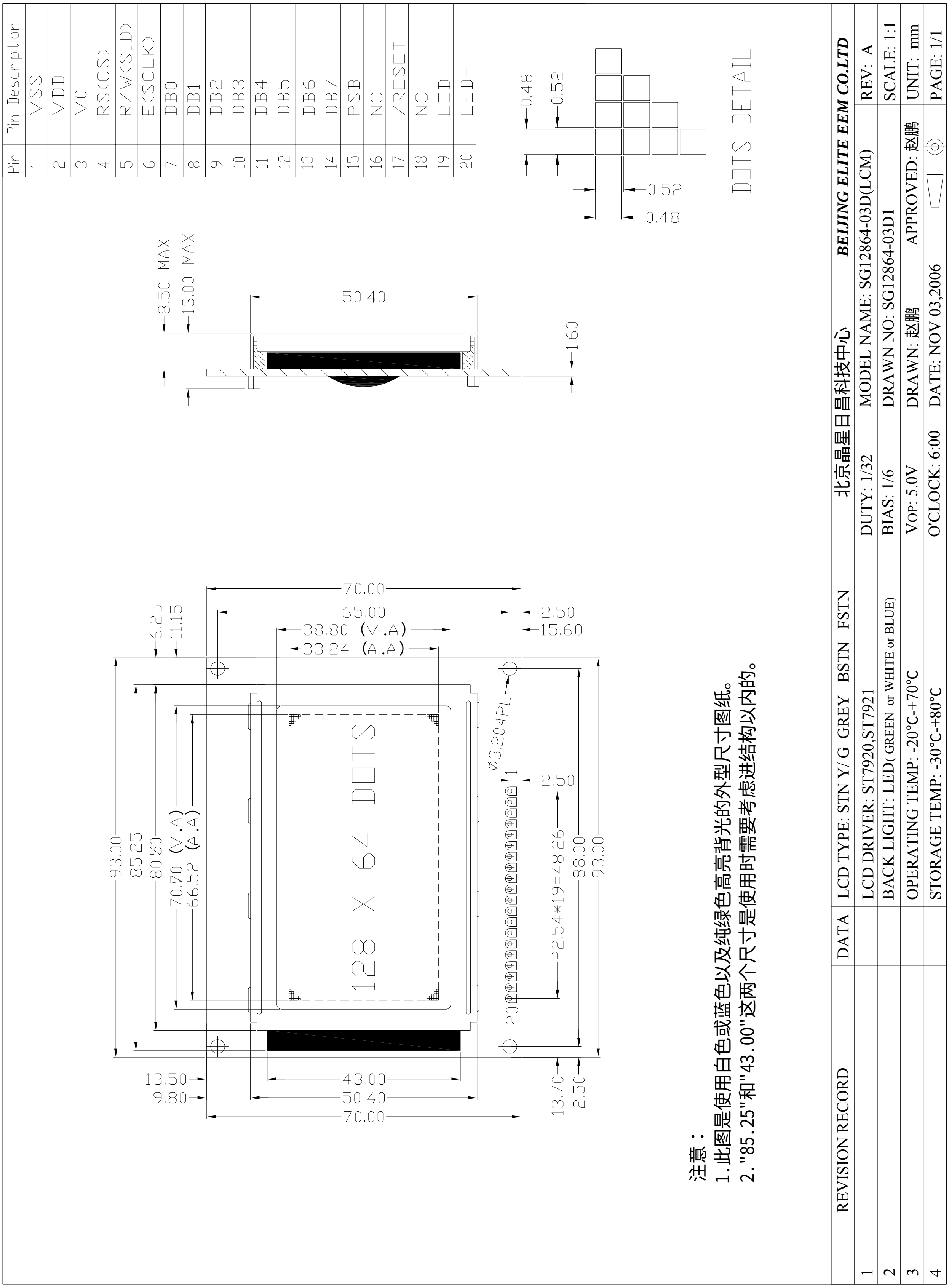
Item	Condition	Standard	Note
High temp. storage	80℃,120 hrs	Appearance without defect	---
Low temp. storage	－30℃,120 hrs	Appearance without defect	---
High temp. operation	70℃,240 hrs	Appearance without defect	---
Low temp. storage	－20℃,240 hrs	Appearance without defect	---
High temp. & humi. storage	50℃,90% RH,120 hrs	Appearance without defect	---
High temp .& humi. operation	40℃,90% RH,120 hrs	Appearance without defect	---
Thermal shock	-20 ℃ ,30min → +25 ℃ ,5min → +60 ℃,30min	Appearance without defect	10 cycles

12. 接口引脚定义

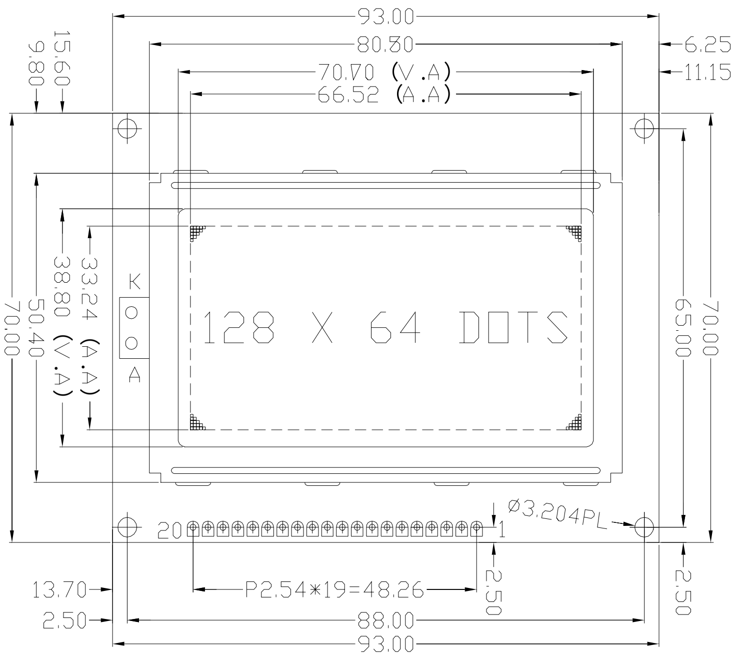
Pin No.	Symbol	Level	Function
1	VSS	0V	电源地
2	VDD	2.7--5.0V	电源输入正
3	VO	---	LCD 驱动电源输入端，本系列的所有模块，此端悬空即可点亮。 后缀为 5V 的模块，可以硬件调节对比度，请参考第 9 项的电路 后缀为 3V 的模块，不可以硬件调节对比度
4	RS（CS）	H/L	并行模式时 高电平：数据，低电平：命令 串行模式时为芯片使能端，高有效
5	R/W(SID)	H/L	并行模式时：高电平：读 ，低电平：写 串行模式时：串行数据端
6	E(SCLK)	H, H → L	并行模式时，LCD 读写信号起始端 串行模式时，串行时钟端
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	PSB	H/L	H:并行模式　　L：串行模式
16	NC	－	厂家保留，必须悬空
17	/RESET	L	低电平复位
18	NC	－	厂家保留，必须悬空
19	LED+	3.0－5.0V	背光电源正
20	LED-	0V	背光电源地

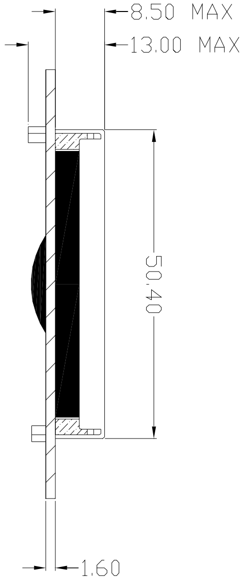
注：请仔细对照产品标签和第二页的列表，以便确定模块的 VDD、LED+的范围，以免误操作损坏模块。

13. 外形尺寸图纸

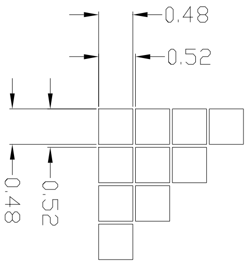


Pin
Pin Description
1 VSS
2 VDD
3 V0
4 RS(CS)
5 R/W(SID)
6 E(SCLK)
7 DB0
8 DB1
9 DB2
10 DB3
11 DB4
12 DB5
13 DB6
14 DB7
15 PSB
16 NC
17 /RESET
18 NC
19 LED+
20 LED-
0.48
0.52
0.52
0.48
DOTS DETAIL
8.50 MAX
13.00 MAX
50.40
1.60
70.00
65.00
38.80 (V.A)
33.24 (A.A)
6.25
11.15
2.50
15.60
Ø3.204PL
2.50
P2.54*19=48.26
88.00
93.00
93.00
85.25
80.30
70.70 (V.A)
66.52 (A.A)
128 X 64 DOTS
1
20
13.50
9.80
43.00
50.40
70.00
13.70
2.50
注意：
1.此图是使用白色或蓝色以及纯绿色高亮背光的外型尺寸图纸。
2."85.25"和"43.00"这两个尺寸是使用时需要考虑进结构以内的。
REVISION RECORD
DATA
LCD TYPE: STN Y/ G GREY BSTN FSTN
LCD DRIVER: ST7920,ST7921
BACK LIGHT: LED(GREEN or WHITE or BLUE)
OPERATING TEMP: -20°C-+70°C
STORAGE TEMP: -30°C-+80°C
北京晶星日昌科技中心
BEIJING ELITE EEM CO.LTD
DUTY: 1/32
BIAS: 1/6
VOP: 5.0V
O'CLOCK: 6:00
MODEL NAME: SG12864-03D(LCM)
DRAWN NO: SG12864-03D1
DRAWN: 赵鹏
DATE: NOV 03,2006
APPROVED: 赵鹏
REV: A
SCALE: 1:1
UNIT: mm
PAGE: 1/1
1
2
3
4





Pin	Pin Description
1	VSS
2	VDD
3	V0
4	RS(CS)
5	R/W(SID)
6	E(SCLK)
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7
15	PSB
16	NC
17	/RESET
18	NC
19	LED+
20	LED-



DOTS DETAIL

注意：
1.此图是使用黄绿底背光和侧背光的外型尺寸图。
2.背光已经完全包在铁框以内。

REVISION RECORD		DATA	LCD TYPE: STN Y/ G GREY BSTN FSTN		BEIJING ELITE EEM CO.LTD		
1			LCD DRIVER: ST7920,ST7921	DUTY: 1/32	MODEL NAME: SG12864-03D(LCM)		REV: A
2			BACK LIGHT: LED(YELLOW GREEN or WHITE or BLUE)	BIAS: 1/6	DRAWN NO: SG12864-03D1		SCALE: 1:1
3			OPERATING TEMP: -20°C-+70°C	VOP: 5.0V	DRAWN: 赵鹏	APPROVED: 赵鹏	UNIT: mm
4			STORAGE TEMP: -30°C-+80°C	O'CLOCK: 6:00	DATE: NOV 03,2006		PAGE: 1/1

14.液晶显示模块在使用过程中的注意事项

14.1 液晶显示模块

▼液晶显示模块在操作过程中的注意事项

我们在出厂前已经针对液晶显示模块进行了精确的装配和调试,因此在客户使用操作时请注意以下几点:

(1) 液晶显示模块避免受到强烈的震动.
(2) 液晶显示模块避免扭动,拆卸金属钮角.
(3) 液晶显示模块避免在印有线路的工作平台上操作.
(4) 除了液晶显示模块的焊盘(输入/输出接线处), 禁止在线路板上的其它地方焊接.
(5) 避免接触,调整,修改导电橡胶.

▼严防静电

液晶显示模块的控制,驱动电路是 CMOS 电路,极易被静电击穿,因此我们在制造和运输整个过程中都采取了严格的防静电措施.请在使用过程中小心,要严防静电,以保持 CMOS IC 的正常工作状态.

(1) 在装配使用液晶显示模块前,请不要将其从包装袋中取出.
液晶显示模块所使用的包装袋是经过防静电处理的特殊包装袋.因此在焊接模块连线之前请不要将其从包装袋中取出.在储存液晶显示模块时也要带有包装袋储存,或者储存在做过防静电处理的容器中,或者放在能充分接地的容器中储存.

(2) 在操作液晶显示模块时,要始终保持操作人充分接地.
将液晶模块从防静电袋里取出时必须保持操作人的充分接地,使人体和液晶模块保持同一电位.从防静电袋里取出的液晶显示模块需要挪动时,应将其放在能充分接地的容器中进行挪动.
此外,操作时应避免穿化学纤维的工作服,最好穿棉的或者经过抗静电处理的工作服.

(3)使用绝缘的,良好接地的电烙铁进行焊接液晶显示模块.
焊接使用的电烙铁必须良好接地,没有漏电.

(4) 在操作过程中所需的设备要充分接地.
在操作液晶显示模块时需要的设备,尤其是驱动器,必须良好接地,没有漏电,以避免干扰.

(5) 使操作台同一电位等于接地.
如果操作台用铝或钢作为接地材料,由于它们抗阻太低,所以可能损坏液晶显示模块或者产生电震. 因此,操作台应使用橡胶垫.

(6) 应慢慢揭去液晶显示模块保护膜.
液晶模块表面都有一层保护膜,目的在于避免造成LCD的偏光片划伤,沾染污渍等.如果快速揭去保护膜都将产生静电,因此要慢慢揭去保护膜.

(7) 注意厂房的湿度
厂房湿度范围: 50~60%RH

▼焊接液晶显示模块时的注意事项:

在焊接液晶显示模块时应注意以下事项:

※ 液晶显示模块上只有输入/输出连线处可以焊接.
※ 焊接所需的烙铁必须绝缘.

(1) 焊接时所需条件:
电铁的温度: 280℃±10℃
焊接时间: ＜3-4S

焊接材料: 低熔点,可充分熔化的焊锡

避免使用融化后易流动的焊锡,因为在焊接时易渗透到液晶显示模块里面,在清理时易对液晶模块造成污染.此外,为了避免焊接时焊锡对液晶显示模块的污染,应在焊接完成后再揭去液晶显示模块的保护膜.

(2) 重复焊接时注意事项:

由于连接线是穿过模块的焊盘与模块焊接的,所以在拆除时需等到焊锡完全熔化时再移动连接线.若焊锡未能完全熔化就用力移动连接线,就极易造成焊盘损坏或脱落.在拆除连接线时最好使用”吸枪”.此外还应注意,重复焊接不得超过 3 次.

▼长时间储存时注意事项:

当液晶显示模块需要长时间储存时,应遵循以下原则:

如果储存方法不当,将影响偏光片的质量,使显示效果不佳;还容易造成焊盘的氧化,不容易焊接.

(1) 储存时尽可能使用出厂时的原包装.
(2) 储存散装的液晶显示模块时,应先装入防静电袋里,封口严密.置放在免受太阳光,日光灯照射的地方储存.
(3) 储存时应保持低湿度,储存温度最佳范围: 0°C~35°C
    储存时应查阅说明书,根据不同模块的最佳储存温度和储存湿度进行储存.

▼关于电流保护装置

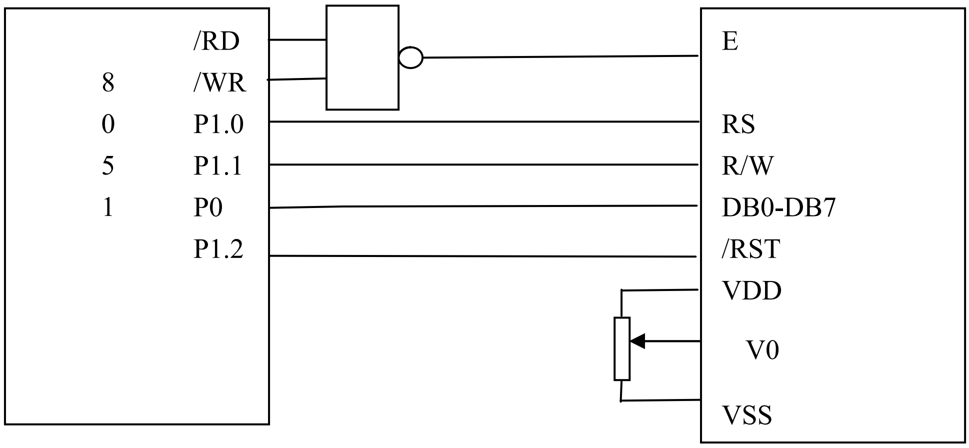
液晶显示模块上没有装电流保护装置,因此,在使用时应预备好电流保护装置.

14.2 液晶显示模块在使用过程中的注意事项

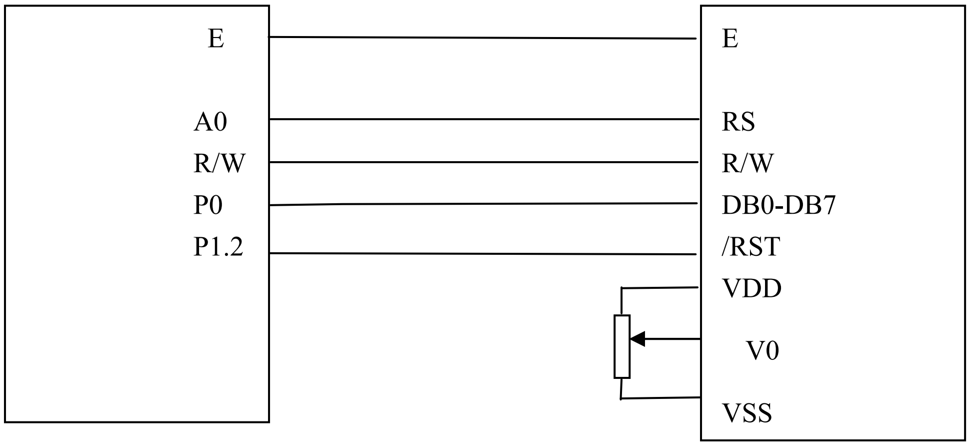
(1) 防止受到振荡,冲击.
(2) 防止用较硬的材料擦拭液晶显示屏表面.
(3) 防止受到挤压.
(4) 防止施加直流电.
(5) 防止太阳光或日光灯的长时间照射.
(6) 避免在高温,高湿度的环境中储存.
(7) 长时间储存时,温度应高于 40°C,湿度应低于 60%.
(8) 液晶显示屏中的液晶材料是有害物质,当不慎溅落到手,身体,衣服等处时,绝对避免入口,应尽快冲洗干净.

15．硬件连接方式（后缀为 5V 的并行模块例）

15.1、非模拟时序方式（直接控制）,80 系列单片机



15.2：68 时序单片机



15.3 模拟时序方式

使用模拟时序访问方式时，将 MPU 的 I/O 口线直接与液晶模块的 I/O 线连接即可。

16．驱动程序

```
/************************************************************************
本驱动程序采用 51 模拟时序方式，晶振 12M，包含两种驱动方式，8 位并行方式
和串行方式。
图片取模方式：横向取模
***********************************************************************/
#include "reg52.h"
#include "intrins.h"
#include "stdio.h"

#define   nop()   _nop_()
#define   lie_value   128
#define   hang_value   64
/***************************************************************************
  *  接口定义及全局变量
 ***************************************************************************/
//serial
sbit      SCLK = P2^6;//串行时钟端
sbit      DI     = P2^7;//串行数据发送端

sbit      REST    = P1^6;//RESET
sbit      A0       = P1^3;//A0
sbit      WR_RW = P1^5;//WRB
sbit      EN_RD = P1^4;//RDB

sbit      KEY     = P3^7;      //KEY
sbit      auto_bit = P3^3;//程序自动手动选择端
sbit      LED_bit   = P3^6;//程序运行指示灯
unsigned char transtype;    //接口类型变量

unsigned char *s;                 //指针变量
bit    flag_user;                 //位变量
unsigned char count;              //定时器计时变量

//延时
void delay(void)
{
      unsigned char i,j;
```

```
	for(i=50;i>0;i--)
		for(j=60;j>0;j--)
		   ;
}
/*************
void delaytt(unsigned int time)
{
	unsigned int i;

    for(i = time;i > 0;i--)
	;
}
*********************/
//检查按键是否有动作,有动作则显示下一屏内容,没动作则等待
void delay_keycheck_A(void)
{
	unsigned int n;

	KEY=1;
	delay();
	delay();
	while(KEY)
	   ;
	for(n=10;n>0;n--)
		delay();
	while(!KEY)
	   ;
}
void delay_keycheck_B(void)
{
	unsigned int i,j,k;

	for(i=0;i<50;i++)
		for(j=0;j<50;j++)
		   for(k=0;k<50;k++)
		      ;
}
void delay_keycheck(void)
{
	if(auto_bit==1)
		delay_keycheck_A();//手动
	else
```

```
		delay_keycheck_B();//自动
}
/***************************************************************************************
   * 7920  ：160X32/122X32/128X64,68/serial
  ***************************************************************************************/
  void busy_check(void)
{ unsigned char key;
	P2=0XFF;
   while(1)
	{
	  A0=0;//命令
	  WR_RW=1;//读
	  EN_RD=1;
	  key=P2;
	EN_RD=0;
	  if((key&0X80)==0X00)
	  break;
      }
}
//写命令:并行 68 时序
void wcom7920_68(unsigned char com)
{
		busy_check();//忙检测
	A0=0;
	WR_RW=0;
	EN_RD=1;
	nop();
	nop();
	nop();
	nop();
	P2=com;
	nop();
	nop();
	nop();
	nop();
	EN_RD=0;
}
//写数据:并行 68 时序
void wdata7920_68(unsigned char dat)
{
busy_check();//忙检测
	A0=1;
	WR_RW=0;
```

```
        EN_RD=1;
        nop();
        nop();
        nop();
        nop();
        P2=dat;
        nop();
        nop();
        nop();
        nop();
        EN_RD=0;
}

//串行发送字节函数
void SendByte7920(unsigned char dat)
{
    unsigned char i;

    A0=1;
    for(i=0;i<8;i++)
    {
        EN_RD=0;
        if(dat&0x80)
                WR_RW=1;
        else
                WR_RW=0;
        EN_RD=1;
        dat=dat<<1;

        nop();
            nop();
    }
}
//写控制命令：串行
void wcom7920_s(unsigned char dat)
{
    SendByte7920(0xF8);//11111,00,0 RW=0,RS=0  同步标志
    SendByte7920(dat&0xF0);//高四位
    SendByte7920((dat&0x0F)<<4);//低四位
}
//写数据命令：串行
void wdata7920_s(unsigned char dat)
{
```

```
    SendByte7920(0xFA);//11111,01,0 RW=0,RS=1
    SendByte7920(dat&0xF0);//高四位
    SendByte7920((dat&0x0F)<<4);//低四位
}
//总的 7920 发送命令函数：68/serial
void wcom7920(unsigned char com)
{
        switch(transtype)
        {
                case 1://并行 68
                                PS=1;
                                wcom7920_68(com);
                                break;
                case 2://串行
                                PS=0;
                                wcom7920_s(com);
                                break;
                default:

                                break;
    }
}
//总的 7920 发送数据函数：68/serial
void wdata7920(unsigned char dat)
{
        switch(transtype)
        {
                case 1://并行 68
                                wdata7920_68(dat);
                                break;
                case 2://串行
                                wdata7920_s(dat);
                                    break;
                default:

                        break;
        }
}
//
void Display7920lines_all_1(void)
{
        unsigned char row;
        unsigned int counta,countb;
```

```
	row=0x80;
	for(counta=32;counta>0;counta--)
	{
		wcom7920(row++);
		wcom7920(0x80);
		for(countb=(lie_value/8);countb>0;countb--)
			wdata7920(0xaa);
	}

		row=0x80;
		for(counta=32;counta>0;counta--)
		{
			wcom7920(row++);
			wcom7920(0x88);
			for(countb=(lie_value/8);countb>0;countb--)
				wdata7920(0xaa);
		}

}
void Display7920lines_all_on(void)
{
	unsigned char row;
	unsigned int counta,countb;

	row=0x80;
	for(counta=32;counta>0;counta--)
	{
		wcom7920(row++);
		wcom7920(0x80);
		for(countb=(lie_value/8);countb>0;countb--)
			wdata7920(0xff);
	}

		row=0x80;
		for(counta=32;counta>0;counta--)
		{
			wcom7920(row++);
			wcom7920(0x88);
			for(countb=(lie_value/8);countb>0;countb--)
				wdata7920(0xff);
		}
```

```
}
void initlcd_char(void)
{
		wcom7920(0x30);
		wcom7920(0x0c);
		wcom7920(0x02);
		wcom7920(0x80);
		wcom7920(0x06);
		wcom7920(0x0c);
}
//printf 函数用到的函数。要在 STDIO.H 中将原有的 PUTCHAR 函数屏蔽。
void putchar(int cc)
{
	unsigned char c;

	c=cc;
	switch(c)
	{
		case '\c' :		//clear
			wcom7920(0x01);
			break ;
		case '\f' :		//first line
			wcom7920(0x80);
			break ;
		case '\s' :		//second line
			wcom7920(0x90);
			break ;
		case '\t' :		//3 line
			wcom7920(0x88);
			break ;
		case '\d' :		//4 line
			wcom7920(0x98);
			break ;
		default :
			wdata7920(c);		//data
			break;
	}
}
void Display7920zi12864(void)
{
	wcom7920(0x30);
	wcom7920(0x01);//clear
	delay();
```

```
        delay();
        initlcd_char();
        printf("\t    北京集粹!");
}
////////////////////////////////////////
void init7920(void)
{

        wcom7920(0x30);
        wcom7920(0x01);//clear
        delay();
        delay();
        wcom7920(0x34);
        wcom7920(0x36);
        wcom7920(0x0c);
        wcom7920(0x06);
        wcom7920(0x03);
        wcom7920(0x40);
        wcom7920(0x02);
        delay();
        delay();
}
/*******************************************************************************
    * main
  ******************************************************************************/
void main(void)
{
        unsigned char tq;
      REST=0;
        delay();
    delay();
    delay();
    delay();
    delay();
        REST=1;

        delay();
    delay();
    delay();
    delay();
    delay();
    delay();
```

```
    EA = 1;                //开总中断
    ET0 = 1;               //允许定时器 0 中断

    TMOD = 1;                //定时器工作方式选择
    TL0 = 0x06;
    TH0 = 0xf8;         //定时器赋予初值

    TR0 = 1;               //启动定时器



    tq=P0 & 0x0f;
        if(tq&0x01)
        transtype=2; //串行
    else
        transtype=1;    //并行
         while(1)
    {
                    init7920();//

                   Display7920lines_all_1(); //显示行 A
        delay_keycheck();

        Display7920zi12864();           //显示汉字
        delay_keycheck();
         }
}

void timer0(void) interrupt 1 using 0
{
    TL0=0x06;
    TH0=0xf8;
    if (++count > 100)
    {
        count = 0;
        LED_bit=~LED_bit;
    }
}
```